UDACITY

< Return to Classroom

DISCUSS ON STUDENT HUB

Part of Speech Tagging

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Dear Learner,
I appreciate and commend the efforts and hard work put into this piece. Congratulations👏👏for making it pass this stage of learning with us and I wish that this spirit is carried forward in subsequent projects. You should be proud of yourself because success is no accident. It is hard work, perseverance, learning, studying, sacrifice and most of all, love of what you are doing or learning to do. Please keep practicing on these projects and I wish you all the best.💪

General Requirements

- Includes `HMM Tagger.ipynb` displaying output for all executed cells
- Includes `HMM Tagger.html`, which is an HTML copy of the notebook showing the output from executing all cells

All files required have been included in this submission. Well done!

Submitted notebook has made no changes to test case assertions

No changes have been made to test case assertions.

Baseline Tagger Implementation

Emission count test case assertions all pass.

- **The emission counts dictionary has 12 keys, one for each of the tags in the universal tagset**
- **"time" is the most common word tagged as a NOUN**

The emission counts dictionary has 12 keys and time is tagged as a NOUN. This is excellent!

Baseline MFC tagger passes all test case assertions and produces the expected accuracy using the universal tagset.

- **>95.5% accuracy on the training sentences**
- **93% accuracy the test sentences**

The Baseline MFC tagger is correctly implemented and all the test case assertions passed. The training and test sentences accuracies are also as expected. Well done!

> training accuracy mfc_model: 95.72%
> testing accuracy mfc_model: 93.01%

Calculating Tag Counts

All unigram test case assertions pass

Unigram test case assertions passed. Well done!

Pro Tips

Here is another pythonic approach to calculating the unigrams.

```
def unigram_counts(sequences):
```

```
return Counter(chain(*sequences))
tag_unigrams = unigram_counts(data.training_set.Y)
```

All bigram test case assertions pass

Bigram test case assertions passed. Well done!

Pro Tips

Here is another pythonic approach to calculating the bigrams.

```
def bigram_counts(sequences):
counts = Counter()
counts.update(chain(*(zip(s[:-1], s[1:]) for s in sequences)))
return counts
tag_bigrams = bigram_counts(data.training_set.Y)
```

All start and end count test case assertions pass

All the start and end count test case assertions passed as expected.

Basic HMM Tagger Implementation

All model topology test case assertions pass

Good job implementing the correct model topology for the Basic HMM Tagger. All its test case assertions passed to confirm its correctness. Well done!👏🏽

Basic HMM tagger passes all assertion test cases and produces the expected accuracy using the universal tagset.

- >97% accuracy on the training sentences
- >95.5% accuracy the test sentences

The Basic HMM tagger successfully was correctly implemented to get impressive accuracies on the training and testing sentences. Excellent results!

> training accuracy basic hmm model: 97.54%
> testing accuracy basic hmm model: 95.95%

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review